Aprendizaje Automático

Introducción a las Redes Neuronales Artificiales

Viviana Cotik

2do cuatrimestre 2020

Redes neuronales artificiales (RNA -RN, ANN, NN-)

Aprendizaje supervisado

- Perceptrón simple
- Redes feedforward multicapa
- ...

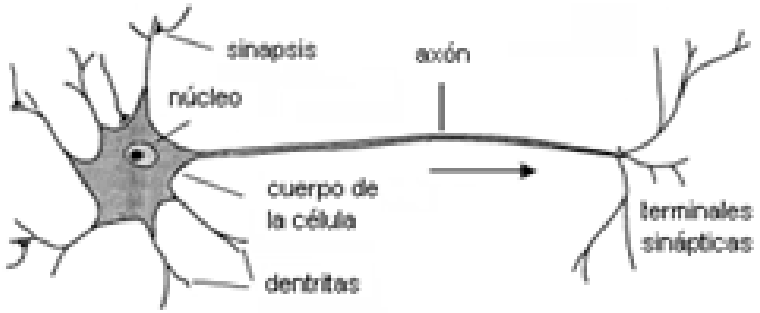
Aprendizaje no supervisado

- Aprendizaje Hebbiano
- Aprendizaje Competitivo
- Mapas Auto-Organizados

Redes neuronales

- inspiradas en un modelo biológico
- conexión de neuronas, unidades de procesamiento sencillas
- opera en paralelo, es robusto ante fallas

La Neurona



Esquema de una neurona

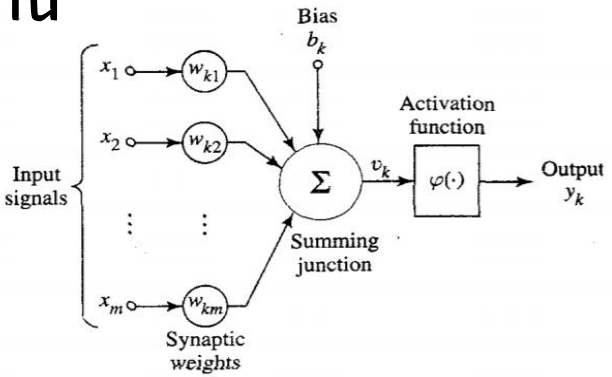
Fuente: Introduction to the theory of Neural Computation. Hertz et al.

procesador de información muy simple:

- Canal de entrada: dendritas.
- Procesador: soma.
- Canal de salida: axón.

Modelo matemático de una neurona

Esquema de una neurona



Esquema del modelo matemático de una neurona – Neural Networks, Haykin.

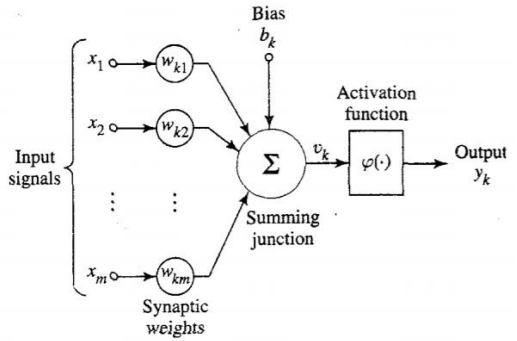
Neurona: **unidad básica de procesamiento de información** de una red neuronal.

Modelo matemático de una neurona dado por:

- **cjto. de conexiones** (sinapsis) con pesos asociados.
- Una **señal xj** conectada a la **neurona k** se multiplica por **peso wkj**
- **sumador de señales** de entrada pesadas por la sinapsis, arroja combinación lineal de las entradas
- una **función de activación** que limita la amplitud de salida de una neurona
- un **umbral bk**, para variar la actividad de la neurona

La neurona

$$O_k = g((\sum_{i=1}^{m} w_{ki} \cdot \xi_i) - b_k),$$



Ej:

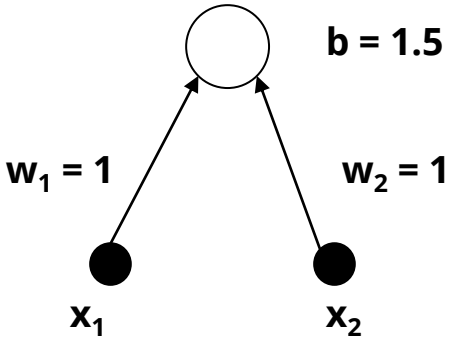
g = sgn($\sum_{i=1}^{m} w_{ki} \cdot \xi_i) - b_k$)

El perceptrón simple resuelve problemas linealmente separables.

Ej. para funciones booleanas: AND, OR, NOT. No funciona para XOR.

Ejemplo: AND

Perceptrón que implementa AND



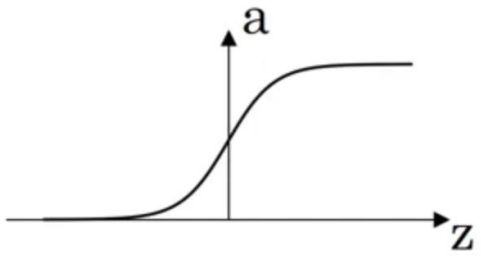
Función de activación: Signo.

En pizarrón:

- Tabla de verdad
- Gráfico de AND

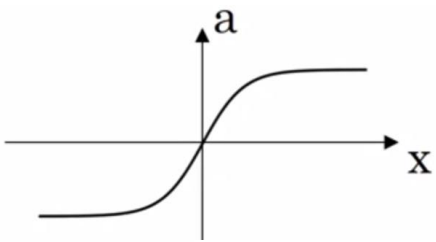
Funciones de activación

Algunas funciones de activación

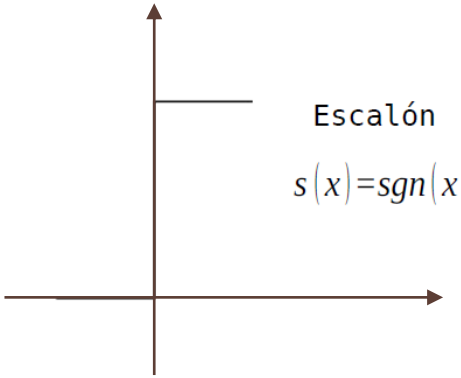


sigmoid: $a = \frac{1}{1+e^{-z}}$

(logística)

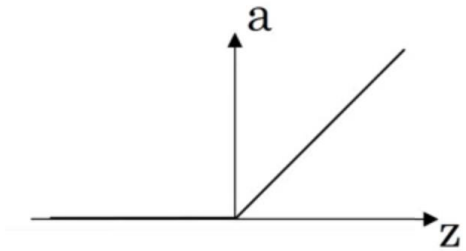


tanh: a = $\frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})}$

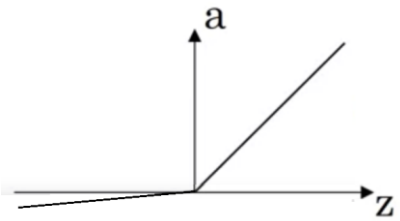


Escalón

$s(x) = sgn(x)$



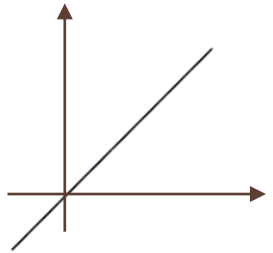
ReLU: a = max (0, z)



leaky ReLU: a = max (0.01 z, z)

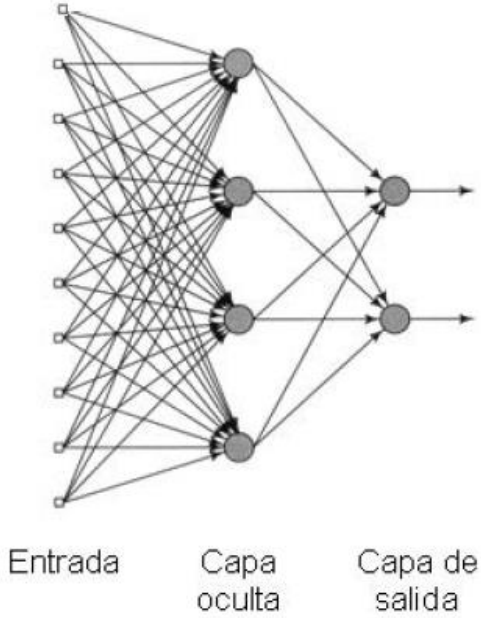
Lineal

$s(x) = x$



Redes neuronales

Arquitectura feed forward

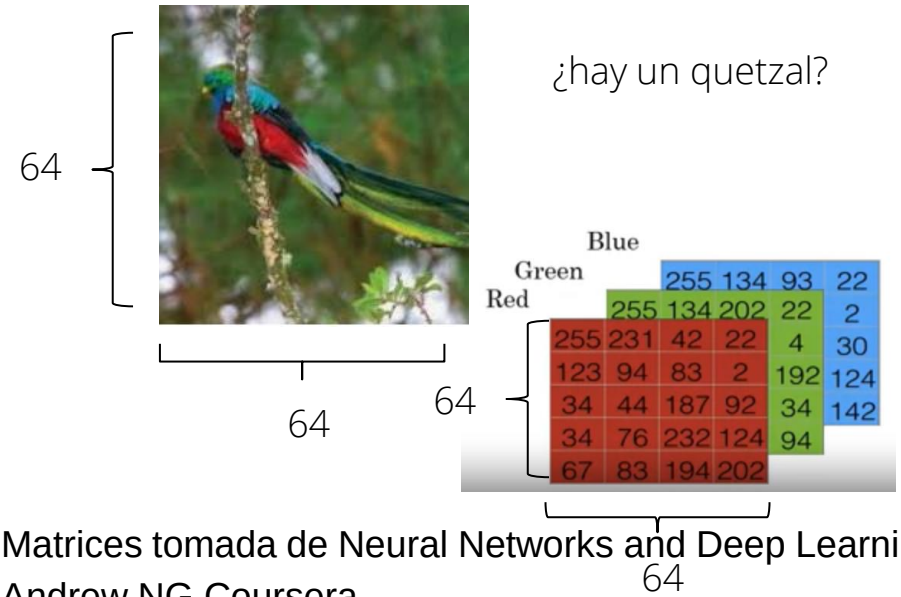


Distintas arquitecturas. Aquí

- perceptrón multicapa
- conexión total

Regresión logística como red neuronal

- algoritmo de clasificación binaria
- representación de imágen en computadora



Matrices tomada de Neural Networks and Deep Learning, Andrew NG Coursera

Entradas

x = 64*64*3 = 12288 (vector de entrada) = n_x

Notación (x,y). x ϵ R^{nx}

y ϵ {0,1}

m= cant. conjuntos de entrenamiento

Regresión logística, funciones de pérdida y de costo

Dado X ϵ R^{nx} , **quiero encontrar $\hat{y}$** = P(y = 1 | x)

$\hat{y} = \sigma(w^T x + b)$ $\sigma(z) = 1/(1+e^{-z})$

Parámetros a aprender: W ϵ R^{nx}, b ϵ R

Función de pérdida:

$$\mathcal{L}(\hat{y}^{(i)}, y^{(i)}) = -y^{(i)} \cdot \log \hat{y}^{(i)} - (1 - y^{(i)}) \cdot \log(1 - \hat{y}^{(i)})$$

Función de costo:

$$J(w,b) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$$

Algoritmo de descenso por gradiente

Teníamos

$$\hat{y} = \sigma(w^T x + b), \ \sigma(z) = \frac{1}{1+e^{-z}}$$

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$

Quiero encontrar w, b que minimicen J(w,b)

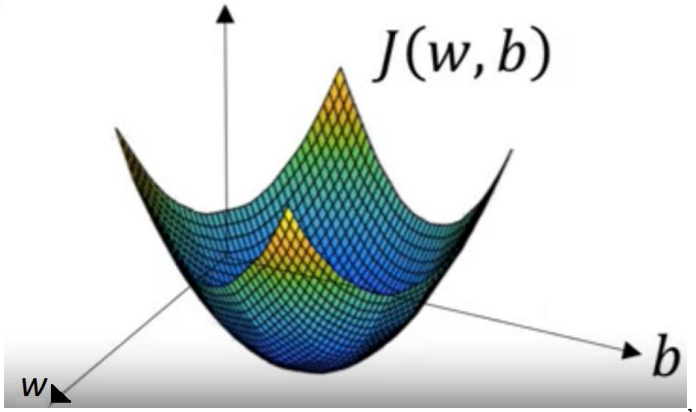
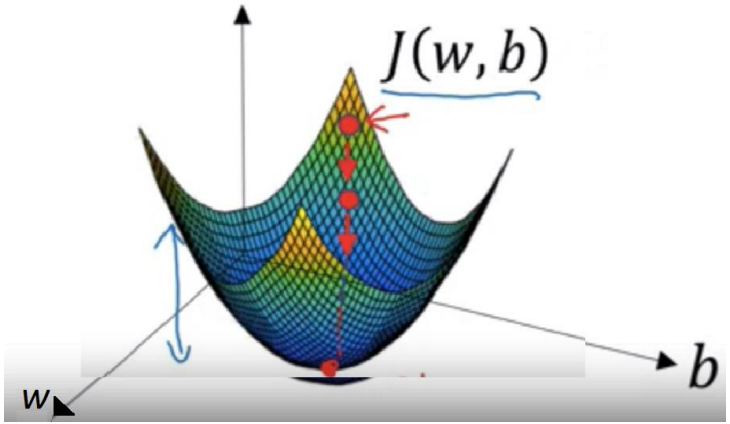


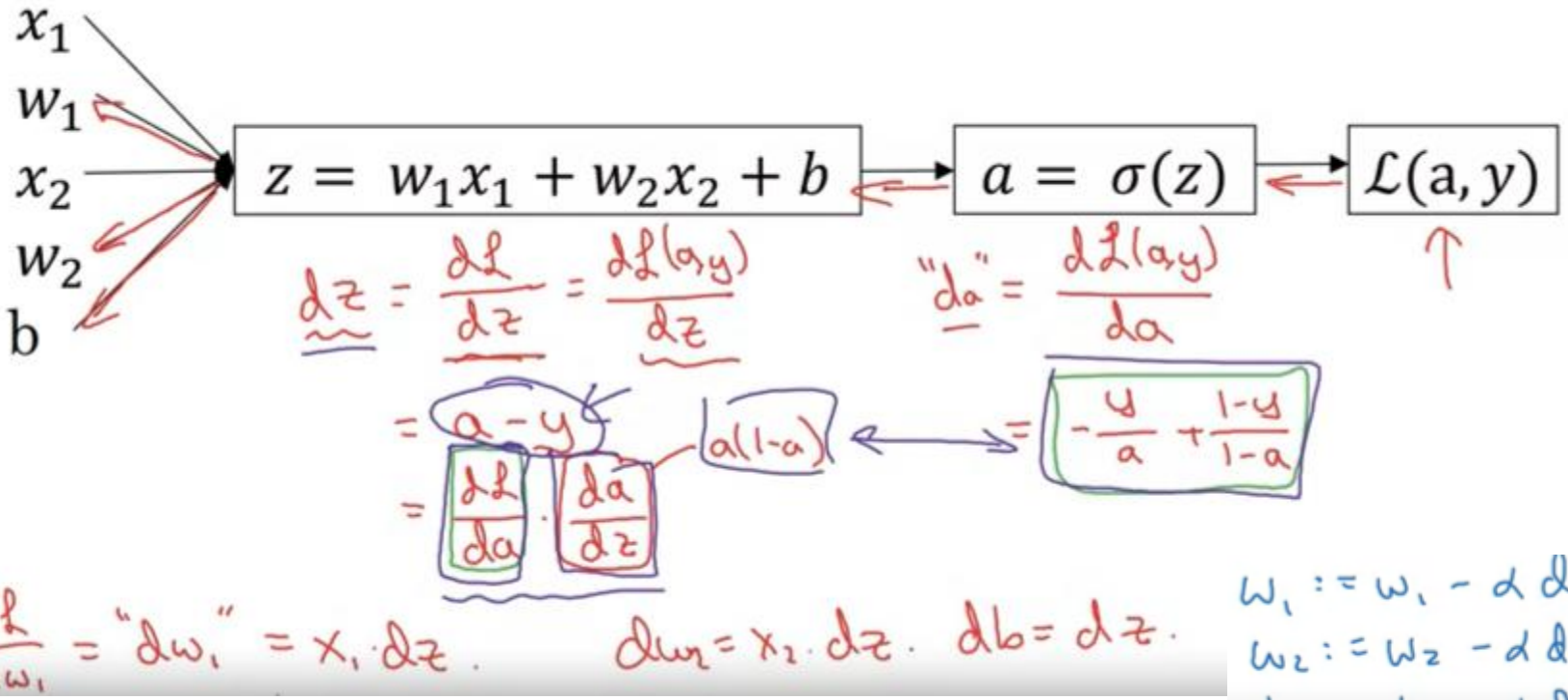
Gráfico tomado de Neural Networks and Deep Learning, Andrew NG Coursera

Algoritmo de descenso por gradiente

$$w \leftarrow w - \eta \frac{\partial J(w,b)}{\partial w}$$

$$b \leftarrow b - \eta \frac{\partial J(w,b)}{\partial b}$$



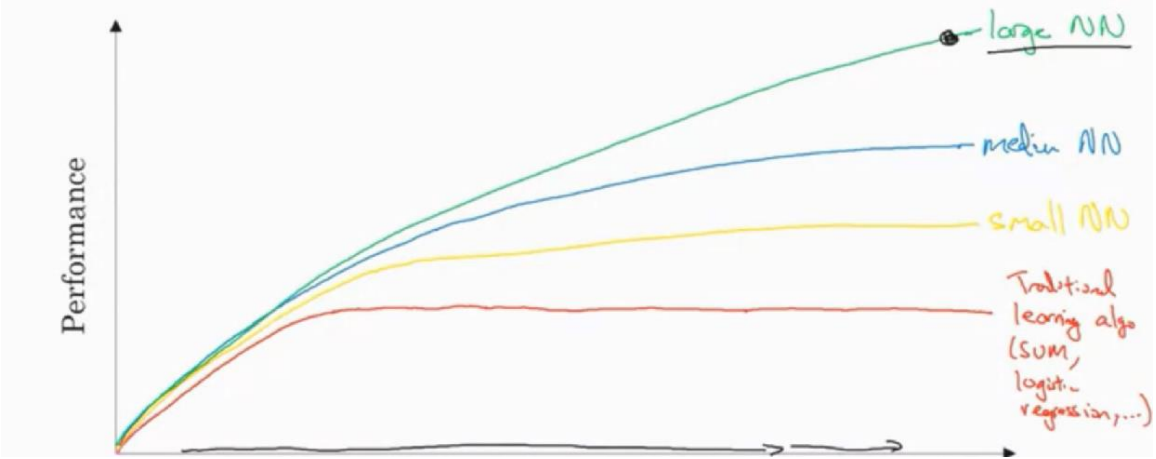


Tomado de Neural Networks and Deep Learning,
Andrew NG Coursera

Arquitecturas de redes neuronales

- Estándares
- Profundas
  - Convolutional Neural Networks (CNN): Imágenes
  - Recurrent Neural Networks (RNN): Texto, Habla
  - Híbridas

Resurgimiento de las RN



- Más **datos** disponibles
- Más poder de **cómputo**
  GPU
- Cambios en **algoritmos**
  ReLU

Tomado de Neural Networks and Deep Learning, Andrew NG Coursera

Parámetros de una red neuronal

- función de activación
- inicialización de pesos
- cantidad de capas ocultas
- conexión entre capas
- tasa de aprendizaje (learning rate)

Hay que evaluarlas en cada caso.

Resumen

Introducción a las redes neuronales artificiales (RN):

- Inspiración biológica
- Modelo matemático. Funciones de activación
- Arquitectura feed-forward. Perceptrón simple y perceptrón multicapa
- Función de pérdida, función de costo
- Método del descenso por gradiente
- Existen variantes de descenso por gradiente para acelerar la convergencia
- Forward y backward pass
- Distintas arquitecturas de RN y resurgimiento de RN
- Poca explicabilidad

Bibliografía

Capítulos de libros:

Mitchell, cap. 4

Libros enteros:

Neural Networks. A comprehensive foundation. Haykin

Introduction To The Theory Of Neural Computation. Hertz, Krogh, Palmer

Deep Learning. Goodfellow, Bengio, Courville

Curso online:

Neural Networks and Deep Learning. Coursera.

Machine Learning. Coursera. Semanas 4 y 5.

Tensor Flow Playground.